ATTORNEY DOCKET NO.: 1004-128

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Nicholas Leavy and Michael L. Hall, Jr.
Serial No.:	10/820,591
For:	USE OF PER-FLOW MONOTONICALLY DECREASING TTLS TO PREVENT IDS CIRCUMVENTION
Filing Date:	April 8, 2004
Examiner:	Choudhury, Azizul Q.
Art Unit:	2445
Conf. No.:	8114

SUBMISSION VIA EFS WEB

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

To the Board:

The information and arguments below are submitted for consideration in this appeal.

Table of Contents

I. Real Party in Interest

The real parties in interest in this matter are Cisco Technology, Inc. (the assignee of record) and Cisco Systems, Inc.

II. Related Appeals and Interferences

There are no related appeal or interference matters.

III. Status of Claims

Claims 1-15, 21-25, 31, 33, 35, and 37 stand finally rejected, as set forth in the final Office Action dated February 8, 2009. Claims 32, 34, 36, and 38 have been withdrawn from consideration. Claims 16-20 and 26-30 are canceled.

The rejection of claims 1-15, 21-25, 31, 33, 35, and 37 is being appealed.

IV. Status of Amendments

There are no after-final amendments pending.

V. Summary of Claimed Subject Matter

Claim 1 recites a method 400 (see Fig. 4) of blocking attacks on a protected computer network 240 (see Fig. 2). The method includes (a) receiving a plurality of packets 250 (see Fig. 2) from a network 220 (see Fig. 2), each packet 250 having a packet time to live (TTL) value 254 (see Fig. 2 and Par. [0017]) and belonging to a corresponding packet flow (see Par. [0024] and step 410 of Fig. 4), (b) storing the smallest packet TTL value 254 (see TTL store 324 of Fig. 3) received from each corresponding packet flow (see Par. [0025] and steps 420, 430, 440 of Fig. 4), and (c) prior to transmitting (step 460 of Fig. 4) each packet 250, setting the packet TTL value

254B (see Fig. 2) to the smallest packet TTL value received (see TTL store 324 of Fig. 3) for the corresponding packet flow (see Par. [0026] and step 450 of Fig. 4).

Claim 6 recites an apparatus 230 (see Fig. 2 and Pars. [0017]-[0019]) for blocking attacks on a protected computer network 240 (see Fig. 2). The apparatus 230 includes means for receiving a plurality of packets 301, 302 (see Fig. 3) from a network 220 (see Fig. 2), each said packet having a packet time to live (TTL) value 254A (see Fig. 2) and belonging to a corresponding packet flow. The apparatus 230 also includes means 350 (see Fig. 3) for storing the smallest packet TTL value 254A received from each said corresponding packet flow (see Par. [0021]), as well as means 330 (see Fig. 3) for setting said packet TTL value 254B to said smallest packet TTL value received for said corresponding packet flow prior to transmitting each said packet 303 (see Fig. 3).

Claim 11 recites an apparatus 230 (see Fig. 2 and Pars. [0017]-[0019]) for blocking attacks on a protected computer network 240 (see Fig. 2). The apparatus 230 includes a packet classifier 310 (see Fig. 3 and Par. [0019]) configured to receive a plurality of packets 301, 302 (see Fig. 3) from a network 220 (see Fig. 2), each said packet having a packet time to live (TTL) value 254A (see Fig. 2) and belonging to a corresponding packet flow. The apparatus 230 also includes a memory 324 (see Fig. 3 AND Par. [0019]) configured to store the smallest packet TTL value 254A received from each said corresponding packet flow (see Par. [0021]), as well as a TTL rewrite unit 330 (see Fig. 3) configured to set said packet TTL value 254B to said smallest packet TTL value received for said corresponding packet flow prior to transmitting each said packet 303 (see Fig. 3).

Claim 21 recites a computer program product including a computer-readable medium having instructions stored thereon that when performed by a computer cause the computer to perform various operations (see Par. [0029]). These operations include (a) receiving a plurality of packets 250 (see Fig. 2) from a network 220 (see Fig. 2), each said packet 250 having a packet time to live (TTL) value 254 (see Fig. 2 and Par.

[0017]) and belonging to a corresponding packet flow (see Par. [0024] and step 410 of Fig. 4), (b) storing the smallest packet TTL value 254 (see TTL store 324 of Fig. 3) received from each said corresponding packet flow (see Par. [0025] and steps 420, 430, 440 of Fig. 4), and (c) prior to transmitting (step 460 of Fig. 4) each said packet 250, setting said packet TTL value 254B (see Fig. 2) to said smallest packet TTL value received (see TTL store 324 of Fig. 3) for said corresponding packet flow (see Par. [0026] and step 450 of Fig. 4).

Claim 31 recites the method of Claim 1, wherein storing (see steps 430 and 440 of Fig. 4) the smallest packet TTL 254A (see Fig. 2) value received from each said corresponding packet flow includes, for each said packet, (a) if that packet 250 (see Fig. 2) is the first packet received from said corresponding packet flow, then storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (Pars. [0019]-[0025]), (b) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than (see step 430 of Fig. 4) the stored smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), then storing (see step 440 of Fig. 4) the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), and (c) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value 254A of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining (see step 435 of Fig. 4) from storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow.

Claim 33 recites the apparatus 230 (see Fig. 2 and Pars. [0017]-[0019]) of Claim 6, wherein said means 350 (see Fig. 3) for storing the smallest packet TTL 254A (see Fig. 2) value received from each said corresponding packet flow includes means for, for each said packet, (a) if that packet 250 (see Fig. 2) is the first packet received from said corresponding packet flow, then storing the packet TTL value 254A of that packet 250

as said smallest packet TTL value received from said corresponding packet flow (Pars. [0019]-[0025]), (b) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than (see step 430 of Fig. 4) the stored smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), then storing (see step 440 of Fig. 4) the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), and (c) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value 254A of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining (see step 435 of Fig. 4) from storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow.

Claim 35 recites the apparatus 230 (see Fig. 2 and Pars. [0017]-[0019]) of Claim 11, further comprising a controller 350 (see Fig. 3). The controller 350 is configured to, for each said packet, (a) if that packet 250 (see Fig. 2) is the first packet received from said corresponding packet flow, then store in memory 324 (see Fig. 3 AND Par. [0019]) the packet TTL value 254A (see Fig. 2) of that packet 250 as said smallest packet TTL value received from said corresponding packet flow(Pars. [0019]-[0025]), (b) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value 254A of that packet 250 is less than (see step 430 of Fig. 4) the stored smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), then store in memory 324 the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), and (c) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value 254A of that packet 250 is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refrain (see step 435 of Fig. 4) from storing in memory the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow.

Claim 37 recites the computer program product of Claim 21 (see Par. [0029]), wherein said instructions for storing the smallest packet TTL value 254A (see Fig. 2) received from each said corresponding packet flow comprise instructions that, when performed by the computer, cause the computer to perform the following operations: (a) if that packet 250 (see Fig. 2) is the first packet received from said corresponding packet flow, then storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow(Pars. [0019]-[0025]), (b) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than (see step 430 of Fig. 4) the stored smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), then storing (see step 440 of Fig. 4) the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), and (c) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value 254A of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining (see step 435 of Fig. 4) from storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow.

VI. Grounds of Rejection to be Reviewed on Appeal

1. Whether claims 1, 6, 11, and 21 are anticipated under 35 U.S.C. § 102(b) by "Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics" (Handley, et al.).

2. Whether claims 31, 33, 35, and 37 are unpatentable under 35 U.S.C. § 103(a) based on the combination of Handley and U.S. Patent Publication No. 2003/00095494 (McElligott).

VII. Argument

1. **Claims 1, 6, 11, and 21 (as well as dependent claims 2-5, 7-10, 12-15, and 22-25) are novel under 35 U.S.C. § 102(b) in view of *Handley* because the cited prior art does not teach "setting said packet TTL value to said smallest packet TTL value received for said corresponding packet flow."**

Claim 1, which is representative of the subject matter of claims 6, 11, and 21, recites a method 400 (see Fig. 4) of blocking attacks on a protected computer network 240 (see Fig. 2). The method includes (a) receiving a plurality of packets 250 (see Fig. 2) from a network 220 (see Fig. 2), each packet 250 having a packet time to live (TTL) value 254 (see Fig. 2 and Par. [0017]) and belonging to a corresponding packet flow (see Par. [0024] and step 410 of Fig. 4), (b) storing the smallest packet TTL value 254 (see TTL store 324 of Fig. 3) received from each corresponding packet flow (see Par. [0025] and steps 420, 430, 440 of Fig. 4), and (c) prior to transmitting (step 460 of Fig. 4) each packet 250, setting the packet TTL value 254B (see Fig. 2) to the smallest packet TTL value received (see TTL store 324 of Fig. 3) for the corresponding packet flow (see Par. [0026] and step 450 of Fig. 4).

The rejections of claims 1, 6, 11, and 21 under 35 USC § 102(b) as being anticipated by Handley are improper because the requirements of that statute are not satisfied.

35 U.S.C. § 102 clearly states that "[a] person shall be entitled to a patent unless" one of seven conditions is satisfied. At issue in this case is the condition set forth in 35 U.S.C. § 102(b) – "the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States."

The meaning of the term "patented or described" has been clarified by the U.S. court of Appeals for the Federal Circuit; "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628,

631, 2 USPQ.2d 1051, 1053 (Fed. Cir. 1987).

The rejection of claim 1 under 35 USC §102(b) as being anticipated by Handley is improper because the cited prior art does not teach "each and every element as set forth in the claim." In particular, Handley does not teach a method which includes "setting said packet TTL value to said smallest packet TTL value received for said corresponding packet flow," as set forth in claim 1.

Handley teaches several methods by which an attacker can evade detection by a network intrusion detection system (NIDS) by exploiting ambiguities (Pages 1-3). In particular, one ambiguity that is discussed includes the situation in which a packet arrives at a NIDS with a value in its time-to-live (TTL) field which is too small to allow it to reach its destination end-system. This is noted to be problematic because the NIDS may then have an incorrect model of the protocol state of the end-system, allowing an attacker to covertly issue malicious commands (Page 2, Col. 1, item iii and Figure 1). Handley then discloses a normalizer which is capable of altering packets to remove certain ambiguities to prevent these kinds of attacks from succeeding (Pages 3-15). In one situation, the Handley normalizer attempts to prevent an attacker from finding a way to systematically ensure that some packets will be received by an end-system of the NIDS and some not. In particular, if a packet arrives at the NIDS with a value in its time-to-live (TTL) field which is too small to allow it to reach the end-system, the Handley normalizer increases the original value in the TTL field to a larger value so that the packet reaches the end-system. Handley calls this larger value the "minimum" since it is just large enough to ensure that the packet does reach the end-system (Page 4, Col. 2, fourth paragraph and Page 9, Col. 1, at TTL solution #3).

The cited prior art does not teach "setting said packet TTL value to said *smallest packet TTL value received* for said corresponding packet flow" (emphasis added). The **final Office Action has cited** a portion of the prior art, which, **upon a cursory glance appears to be relevant**. However, upon anything more than such a cursory glance, it is **clear that the cited portion is completely different** than the claimed subject matter.

The Office Action, on page 3, cites page 9, left column, TTL solution #3 of Handley as teaching this feature. That cited portion states:

> Configure the normalizer with a TTL that is larger than the longest path across the internal site. If packets arrive that have a TTL lower than the configured minimum, then the normalizer restores the TTL to the minimum.

Although the word "minimum" is (perhaps misleadingly) used, contrary to the assertion of the Office Action, this cited portion does not teach "setting said packet TTL value to said *smallest packet TTL value received* for said corresponding packet flow" (emphasis added) for at least the following three reasons:

a. The method taught by *Handley* is inconsistent with the method of claim 1 because in claim 1, the setting has to be a decrease (based upon the language of the other limitations), while in *Handley*, the restoring of the TTL to the minimum inherently must be an increase.

In claim 1, each packet "belong[s] to a corresponding packet flow." Thus, inherent in the claim is the truth that for each packet being processed, the "smallest packet TTL value received from each said corresponding packet flow" cannot possibly be any larger than the TTL of that packet (since, employing *reductio ad absurdum* logic, if the TTL of that packet were smaller than the smallest received value from that packet flow, then the smallest received value would no longer actually be the smallest TTL value received from that packet flow because the TTL of that packet would be smaller). Thus, "setting said packet TTL value to said *smallest packet TTL value received* for said corresponding packet flow" **can only result in the TTL value of that packet being maintained or decreased**.

However, in Handley, "**[i]f** packets arrive that **have a TTL lower** than the configured minimum, then the normalizer **restores the TTL to the minimum**." Thus, if a packet, upon being processed, has a TTL greater than or equal to the configured minimum, no action is taken (or at least, Handley does not teach that any action is to be

taken). **Only if the packet being processed has a TTL less than the configured minimum** does Handley teach that the normalizer alters the TTL of the packet to the minimum allowable value. Such **alteration then inherently takes the form of increasing the TTL** of the processed packet.

Thus, it would be **impossible** to perform the method of claim 1 in a way that would be anticipated by the cited portions of Handley, since the claimed element inherently requires the packet TTL value to be **decreased** under certain conditions, while the method taught by Handley inherently requires the packet TTL value to be **increased**.

b. Restoring the TTL to the minimum is not equivalent to setting said packet TTL value to said smallest packet TTL value received for said corresponding packet flow, contrary to the assertion of the Office Action.

The Office Action, on page 7, argues that restoring the TTL to the minimum (as taught by Handley) is equivalent to setting said packet TTL value to said *smallest packet TTL value received* for said corresponding packet flow (as recited in claim 1). This assertion clearly results from a **misunderstanding of the language of Handley**.

Thus, in order to clarify the issue, the meaning of the term "minimum" in the phrase "the normalizer restores the TTL to the minimum" must be construed. Looking back at the previous sentence of Handley, the reader is instructed to "[c]onfigure the normalizer with a TTL that is larger than the longest path across the internal site." Page 9, Col. 1, at TTL solution #3). Thus, the normalizer is configured with a minimum allowable TTL for an incoming packet, such minimum being "larger than the longest path across the internal site." The next sentence of Handley makes clear that if any "packets arrive that have a TTL lower than the configured minimum [i.e, the minimum allowable TTL for an incoming packet as discussed in the previous sentence], then the normalizer restores the TTL [of the packet that has just arrived at the beginning of this sentence] to the minimum [i.e, the minimum allowable TTL for an incoming packet as

discussed in the previous sentence]." Thus, a minimum TTL is set and any packet that arrives with a TTL less than the minimum TTL is altered to have the minimum allowable TTL value. This is the only way that a person having ordinary skill in the art could interpret the cited portion.

The "smallest packet TTL value received from each said corresponding packet flow" of claim 1 is not analogous or equivalent to the configured minimum path across the internal site of Handley. Rather, in claim 1, clearly, the *smallest packet TTL value received* is the smallest TTL value of any packet received in a packet flow, which depends on all packets received (and analyzed) up to the packet being processed. On the other hand, the configured minimum of Handley does not depend on any packets previously received – it is instead calculated based on the length of "the longest path across the internal site," which is **invariant**. Thus, the "smallest packet TTL value received from each said corresponding packet flow" of claim 1 is **not analogous or otherwise equivalent** to the configured minimum path across the internal site of Handley. Thus, restoring the TTL to the minimum is **not equivalent** to setting said packet TTL value to said *smallest packet TTL value received* for said corresponding packet flow.

c. The method of claim 1 and the method of *Handley* operate according to entirely different principles.

Handley prevents any packet with too small of a TTL from expiring before reaching its destination within a network. Thus, **the network is protected from attacks due to malicious retransmission packets because no retransmission packets will ever be needed**.

On the other hand, the method of claim 1 allows packets with a TTL that is too small to pass by, but since those packets will not be received by the end host, **retransmission packets will be needed, so the method of claim 1 ensures that any retransmission packets will not have a TTL that is higher than that of the original**

packet in order to prevent the retransmission packets from effectuating any harm (since the retransmission packet will not traverse the network any farther than the original packet).

Thus, the method of claim 1 and the method of Handley clearly operate according to entirely different principles.

Thus, Handley does not teach "setting said packet TTL value to said *smallest packet TTL value received* for said corresponding packet flow," and, therefore, 35 U.S.C. § 102(b) does not preclude the patentability of claim 1. Thus, the rejection of claim 1 under 35 U.S.C. § 102(b) should be removed. Because claims 6, 11, and 21 recite limitations similar to the limitations recited in claim 1, the rejections of claims 6, 11, and 21 under 35 U.S.C. § 102(b) should also be removed for similar reasons. Because claims 2-5, 7-10, 12-15, and 22-25 depend from claims 1, 6, 11, and 21, respectively, the rejections of claims 2-5, 7-10, 12-15, and 22-25 under 35 U.S.C. § 102(b) should also be removed for similar reasons. Thus, claims 1-15 and 21-25 are allowable.

2. Claims 31, 33, 35, and 37 are patentable under 35 U.S.C. § 103(a) in view of the combination of *Handley* and *McElligott*.

Claim 31, which is representative of the subject matter of claims 33, 35, and 37, recites the method of Claim 1, wherein storing (see steps 430 and 440 of Fig. 4) the smallest packet TTL 254A (see Fig. 2) value received from each said corresponding packet flow includes, for each said packet, (a) if that packet 250 (see Fig. 2) is the first packet received from said corresponding packet flow, then storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (Pars. [0019]-[0025]), (b) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than (see step 430 of Fig. 4) the stored smallest packet TTL value

received from said corresponding packet flow (see Par. 0025]), then storing (see step 440 of Fig. 4) the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow (see Par. 0025]), and (c) if that packet 250 is not the first packet received from said corresponding packet flow and the packet TTL value 254A of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining (see step 435 of Fig. 4) from storing the packet TTL value 254A of that packet 250 as said smallest packet TTL value received from said corresponding packet flow.

The rejections of claims 31, 33, 35, and 37 under 35 USC § 103(a) as being anticipated by Handley are improper because the requirements of that statute are not satisfied.

35 U.S.C. § 103(a) clearly states that "[a] patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made."

The rejection of claim 31 under 35 USC § 103(a) as being unpatentable over Handley in view of McElligott is improper because the combination of those two references would not render "the subject matter as a whole [of the claim] . . . obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains."

At the very least, this combination is not obvious, because the two references are directed to very different problems with very different solutions. Furthermore, the rationale provided by the Office Action to combine them does not appear to have any rational basis.

Handley is directed towards techniques for preventing attacks on a network

through the use of a normalizer. McElligot, on the other hand, is directed to techniques for identifying the geographical location of a device (Abstract). These fields are unrelated, and it is unclear why any person having ordinary skill in the art would be motivated to combine these references. The Office Action, on page 6, presents an argument as to why a person having ordinary skill in the art would have combined Handley with McElligot, however, that argument does not actually explain why a person having ordinary skill in the art would be motivated to combine the references. Rather, it merely explains that McElligot teaches **how** to determine if a TTL is lower than a stored value. However, because preventing attacks on a network is completely unrelated to identifying the geographical location of a device, there is no motivation to combine these references. Moreover, no other explanation as to why it would be obvious has been provided. Thus, it would not have been obvious to a person having ordinary skill in the art at the time of the invention to have combined Handley with McElligot.

Thus, it would not have been obvious to a person having ordinary skill in the art to combine Handley with McElligot, and, therefore, 35 U.S.C. § 103(a) does not preclude the patentability of claim 31. Thus, the rejection of claim 31 under 35 U.S.C. § 103(a) should be removed. Because claims 33, 35, and 37 recite limitations similar to the limitations recited in claim 31, the rejections of claims 33, 35, and 37 under 35 U.S.C. § 103(a) should also be removed for similar reasons. Thus, claims 31, 33, 35, and 37 are allowable.

Thus, for the reasons stated above, Applicants respectfully request that the Board set aside the rejections of claims 1-15, 21-25, 31, 33, 35, and 37 and remand the Application back to the Examiner for further consideration.

Respectfully submitted,

/Michael Ari Behar/
M. Ari Behar, Esq.
Attorney for Assignee
USPTO Registration No.: **58,203**
Bainwood, Huang & Associates, LLC
Highpoint Center
2 Connector Road
Westborough, Massachusetts 01581
Telephone: (508) 616-2900
Facsimile: (508) 366-4688

Attorney Docket No.: 1004-128

Dated: July 24, 2009

VIII. Claims Appendix

1. (Original) A method of blocking attacks on a protected computer network, comprising:

receiving a plurality of packets from a network, each said packet having a packet time to live (TTL) value and belonging to a corresponding packet flow;

storing the smallest packet TTL value received from each said corresponding packet flow; and

prior to transmitting each said packet, setting said packet TTL value to said smallest packet TTL value received for said corresponding packet flow.

2. (Previously Presented) The method of Claim 1, wherein said storing the smallest packet TTL value comprises:

associating an epoch with said stored smallest packet TTL value; and

if said epoch is greater than a predefined value, discarding said stored smallest packet TTL value.

3. (Original) The method of Claim 1, further comprising periodically resetting said stored smallest packet TTL value to a maximum value.

4. (Original) The method of Claim 1, wherein said setting said packet TTL value comprises:

determining if said corresponding packet flow is on an unrestricted list;

and if said corresponding packet flow is on said unrestricted list, setting said packet TTL value to a maximum value.

5. (Original) The method of Claim 1, wherein said setting said packet TTL value comprises:

determining if said corresponding packet flow is on an unrestricted list;

and if said corresponding packet flow is on said unrestricted list, leaving said packet TTL value unchanged.

6. (Original) An apparatus for blocking attacks on a protected computer network, comprising:

means for receiving a plurality of packets from a network, each said packet having a packet time to live (TTL) value and belonging to a corresponding packet flow;

means for storing the smallest packet TTL value received from each said corresponding packet flow; and

means for setting said packet TTL value to said smallest packet TTL value received for said corresponding packet flow prior to transmitting each said packet.

7. (Previously Presented) The apparatus of Claim 6, wherein said means for storing the smallest packet TTL value comprises:

means for associating an epoch with said stored smallest packet TTL value; and

means for discarding said stored smallest packet TTL value if said epoch is greater than a predefined value.

8. (Original) The apparatus of Claim 6, further comprising means for periodically resetting said stored smallest packet TTL value to a maximum value.

9. (Original) The apparatus of Claim 6, wherein said means for setting said packet TTL value comprises:

means for determining if said corresponding packet flow is on an unrestricted list; and

means for setting said packet TTL value to a maximum value if said corresponding packet flow is on said unrestricted list.

10. (Original) The apparatus of Claim 6, wherein said means for setting said packet TTL value comprises:

means for determining if said corresponding packet flow is on an unrestricted list; and

means for leaving said packet TTL value unchanged if said corresponding packet flow is on said unrestricted list.

11. (Original) An apparatus for blocking attacks on a protected computer network, comprising:

a packet classifier configured to receive a plurality of packets from a network, each said packet having a packet time to live (TTL) value and belonging to a corresponding packet flow;

a memory configured to store the smallest packet TTL value received from each said corresponding packet flow; and

a TTL rewrite unit configured to set said packet TTL value to said smallest packet TTL value received for said corresponding packet flow prior to transmitting each said packet.

12. (Previously Presented) The apparatus of Claim 11, wherein said memory comprises:

first control means for associating an epoch with said stored smallest packet TTL value; and

second control means for discarding said stored smallest packet TTL value if said epoch is greater than a predefined value.

13. (Original) The apparatus of Claim 11, further comprising control means for periodically resetting said stored smallest packet TTL value to a maximum value.

14. (Original) The apparatus of Claim 11, wherein said TTL rewrite unit comprises:

first control means for determining if said corresponding packet flow is on an unrestricted list; and

second control means for setting said packet TTL value to a maximum value if said corresponding packet flow is on said unrestricted list.

15. (Original) The apparatus of Claim 11, wherein said TTL rewrite unit comprises:

first control means for determining if said corresponding packet flow is on an unrestricted list; and

second control means for leaving said packet TTL value unchanged if said corresponding packet flow is on said unrestricted list.

Claims 16-20 (Canceled).

21. (Previously Presented) A computer program product comprising a computer-readable medium having instructions stored thereon that, when performed by a computer, cause the computer to perform the following operations:

receiving a plurality of packets from a network, each said packet having a packet time to live (TTL) value and belonging to a corresponding packet flow;

storing the smallest packet TTL value received from each said corresponding packet flow; and

prior to transmitting each said packet, setting said packet TTL value to said smallest packet TTL value received for said corresponding packet flow.

22. (Previously Presented) The computer program product of Claim 21, wherein said instructions for storing the smallest packet TTL value comprise instructions that, when performed by the computer, cause the computer to perform the following operations:

associating an epoch with said stored smallest packet TTL value; and

if said epoch is greater than a predefined value, discarding said stored smallest packet TTL value.

23. (Previously Presented) The computer program product of Claim 21, further comprising instructions that, when performed by the computer, further cause the computer to perform the following operations:

periodically resetting said stored smallest packet TTL value to a maximum value.

24. (Previously Presented) The computer program product of Claim 21, wherein said instructions for setting said packet TTL value comprise instructions that, when performed by the computer, cause the computer to perform the following operations:

determining if said corresponding packet flow is on an unrestricted list; and

if said corresponding packet flow is on said unrestricted list, setting said packet TTL value to a maximum value.

25. (Previously Presented) The computer program product of Claim 21, wherein said instructions for setting said packet TTL value comprise instructions that, when performed by the computer, cause the computer to perform the following operations:

determining if said corresponding packet flow is on an unrestricted list; and

if said corresponding packet flow is on said unrestricted list, leaving said packet TTL value unchanged.

Claims 26-30 (Canceled).

31. (Previously Presented) The method of Claim 1, wherein storing the smallest packet TTL value received from each said corresponding packet flow includes, for each said packet:

if that packet is the first packet received from said corresponding packet flow, then storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow;

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than the stored smallest

packet TTL value received from said corresponding packet flow, then storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining from storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow.

32. (Withdrawn) The method of Claim 1, wherein storing the smallest packet TTL value received from each said corresponding packet flow includes, for each said packet:

if that packet is the first packet received from said corresponding packet flow, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if the packet TTL value of that packet is less than or equal to the stored smallest packet TTL value received from said corresponding packet flow, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if an amount of time elapsed since the time indicated by the timestamp is greater than a predefined value, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise:

refraining from storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

refraining from associating a new timestamp with said stored smallest packet TTL value.

33. (Previously Presented) The apparatus of Claim 6, wherein said means for storing the smallest packet TTL value received from each said corresponding packet flow includes means for, for each said packet:

if that packet is the first packet received from said corresponding packet flow, then storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow;

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than the stored smallest packet TTL value received from said corresponding packet flow, then storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining from storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow.

34. (Withdrawn) The apparatus of Claim 6, wherein said means for storing the smallest packet TTL value received from each said corresponding packet flow includes means for, for each said packet:

if that packet is the first packet received from said corresponding packet flow, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if the packet TTL value of that packet is less than or equal to the stored smallest packet TTL value received from said corresponding packet flow, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if an amount of time elapsed since the time indicated by the timestamp is greater than a predefined value, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise:

refraining from storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

refraining from associating a new timestamp with said stored smallest packet TTL value.

35. (Previously Presented) The apparatus of Claim 11, further comprising a controller, the controller being configured to, for each said packet:

if that packet is the first packet received from said corresponding packet flow, then store in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow;

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than the stored smallest packet TTL value received from said corresponding packet flow, then store in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refrain from storing in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow.

36. (Withdrawn) The apparatus of Claim 11, further comprising a controller, the controller being configured to, for each said packet:

if that packet is the first packet received from said corresponding packet flow, then:

store in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associate a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if the packet TTL value of that packet is less than or equal to the stored smallest packet TTL value received from said corresponding packet flow, then:

store in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associate a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if an amount of time elapsed since the time indicated by the timestamp is greater than a predefined value, then:

store in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associate a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, refrain from:

storing in memory the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a new timestamp with said stored smallest packet TTL value.

37. (Previously Presented) The computer program product of Claim 21, wherein said instructions for storing the smallest packet TTL value received from each said corresponding packet flow comprise instructions that, when performed by the computer, cause the computer to perform the following operations:

if that packet is the first packet received from said corresponding packet flow, then storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow;

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is less than the stored smallest

packet TTL value received from said corresponding packet flow, then storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

if that packet is not the first packet received from said corresponding packet flow and the packet TTL value of that packet is greater than the stored smallest packet TTL value received from said corresponding packet flow, then refraining from storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow.

38. (Withdrawn) The computer program product of Claim 21, wherein said instructions for storing the smallest packet TTL value received from each said corresponding packet flow comprise instructions that, when performed by the computer, cause the computer to perform the following operations:

if that packet is the first packet received from said corresponding packet flow, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if the packet TTL value of that packet is less than or equal to the stored smallest packet TTL value received from said corresponding packet flow, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise, if an amount of time elapsed since the time indicated by the timestamp is greater than a predefined value, then:

storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

associating a timestamp with said stored smallest packet TTL value, the timestamp indicating the time at which said smallest packet TTL value received from said corresponding packet flow was stored;

otherwise:

refraining from storing the packet TTL value of that packet as said smallest packet TTL value received from said corresponding packet flow; and

refraining from associating a new timestamp with said stored smallest packet TTL value.

IX. Evidence Appendix

No evidence is cited.

X. Related Proceedings Appendix

None.